

Wire Arc Additive Manufacturing: A Progressive Multi-Architecture Computer Vision Framework for Real-Time Defect Detection

Soham Agarwal

Indian Institute of Technology Kharagpur

Shahana Bano

s.bano@rgu.ac.uk

Robert Gordon University

Anil Prathuru

Robert Gordon University

Somasundar Kannan

Robert Gordon University

Deepika Nikam

Ulster University, Derry-Londonderry Campus

Shiva Sekar

Indian Institute of Technology Jammu

Sagar Nikam

Ulster University, Derry-Londonderry Campus

Research Article

Keywords: WAAM, Computer Vision, Machine Learning, Deep Learning, CNN-LSTM, Learnable Preprocessing, Feature Engineering

Posted Date: July 22nd, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-10373995/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

Wire Arc Additive Manufacturing: A Progressive Multi-Architecture Computer Vision Framework for Real-Time Defect Detection

Soham Agarwal¹, Shahana Bano^{2*}, Anil Prathuru^{2*},
Somasundar Kannan², Deepika Nikam³, Shiva Sekar⁴,
Sagar Nikam³

¹Dept. of Chemical Engineering, Indian Institute of Technology
Kharagpur, Kharagpur, India.

²School of Computing, Engineering and Technology, Robert Gordon
University, Garthdee Road, Aberdeen, AB10 7GJ, United Kingdom.

³Intelligent Systems Research Centre, Ulster University,
Derry/Londonderry, BT48 7JL, United Kingdom.

⁴Laboratory for Advanced Manufacturing and Processing, Indian
Institute of Technology Jammu, Jammu, India.

*Corresponding author(s). E-mail(s): s.bano@rgu.ac.uk;
a.prathuru@rgu.ac.uk;

Abstract

Wire Arc Additive Manufacturing (WAAM) involves complex, multi-parameter interactions in which weld pool stability, arc behavior, and deposition quality must be continuously monitored. To address the limitations of traditional black-box deep learning models, this paper evaluates a progressive, multi-tiered computer vision and machine learning framework. We first establish a highly interpretable two-step architecture utilizing Convolutional Neural Networks (CNNs) for spatial feature extraction from preprocessed video frames, followed by classical machine learning classifiers to establish baseline operational states. Building upon these spatial embeddings, we evaluate end-to-end deep learning models including MobileNetV2 and EfficientNet-B0, and introduce a hybrid Recurrent Convolutional architecture (CNN-LSTM) to capture critical spatial-temporal dynamics. Finally, we propose **MCFE-Net** (Multi-Channel Feature Engineering Network), a novel architecture in which the preprocessing pipeline itself becomes a set of differentiable, learnable modules jointly optimized

with the classification objective. This progressive evaluation explicitly characterizes the trade-offs between computational efficiency, interpretability, and the system’s ability to accurately detect dynamic anomalies and irregular fusion states in real-time WAAM monitoring.

Keywords: WAAM, Computer Vision, Machine Learning, Deep Learning, CNN-LSTM, Learnable Preprocessing, Feature Engineering

1 Introduction

Wire Arc Additive Manufacturing (WAAM) is a prominent directed energy deposition (DED) technology that enables the cost-effective fabrication of large-scale metallic components by depositing material layer-by-layer using an electric arc heat source and metal wire feedstock [2]. Due to the intense thermal gradients, rapid solidification rates, and highly dynamic arc-wire interactions, WAAM processes are highly sensitive to small environmental and operational variations. Minor fluctuations in welding parameters—such as arc current, voltage, torch travel speed, and wire feed rate—can induce severe defect states, including humping, bead waviness, lack of fusion, undercutting, and excessive spatter ejection. Detecting these operational anomalies in real-time is crucial for closed-loop quality control and autonomous parameter correction in safety-critical industrial applications, motivating the development of computer-vision-based monitoring frameworks [11].

To monitor the process, cameras are typically aligned with the welding torch. However, the extreme luminosity of the arc plasma generates intense glare that overexposes the image center, while the solidifying weld bead in the cooling trail is frequently underexposed. Consequently, extraction of clean, discriminative visual features of the weld pool geometry is highly challenging. Traditional process monitoring involving acquiring temporal variations of the current and voltage signals are not adequate to detect process variations. To address this issue recent studies pursued monitoring additional data such as camera-based video capture, microphone-based audio recordings etc. However, several of these studies are aimed at monitoring a single physical phenomenon.

For instance, Kim et al [3] used SVMs to classify the bead central angle as ‘Good’ and ‘Bad’ based on the process parameter combinations. They considered 20 deposition conditions for the model training followed by 8 randomly selected experimental conditions for model verification. They demonstrated the effectiveness of SVMs in quality classification even with the limited amount of data considered. In addition, they also showed that the developed classifier is effective when multilayer deposition is considered. Distance between the weld tip (nozzle) and the work surface plays a significant role in material deposition rate. Undulance in the previously deposited layer introduce cumulative variations over multiple deposited layers. Franke et al [4] used camera-based image processing to monitor the nozzle to work surface distance to enable closed loop control. This parameter is correlated with spatter formation. A spatter monitoring algorithm was developed along with a 3-stage wire information

extraction approach. The first stage had a neural network to segment and locate the wire followed by a classification to check the integrity and validation for final feature extraction.

Research on combining multiple data sources to create a single process monitoring methodology was also carried out, microphone-based audio recordings to detect variations in arc due to fluctuations in the current and voltage characteristics being one of the prominent ones. For instance, Clark et al [5] optimised the parameters of a convolutional neural network (CNN) with a multimodal waveform-based input comprised of current, voltage and microphone data with varying sampling rates. The collected data from the 3 different sources was windowed with varying window time lengths and overlap lengths to create a single concatenated vector which was then used as the training data to detect defective specimens. Defects namely weld discontinuity, porosity, crack/delamination and poor adhesion have been considered. Defect prediction accuracies of higher than 95% were demonstrated. Kim et al [6] proposed an process image capture methodology where in video recordings were captured from multiple viewpoints/angles to create the training dataset. This is to account for variabilities within the segmentation training image datasets used even for a single set of process control parameters. WAAM videos were captured at multiple camera orientations and correlated with the weld bead geometry captured using a laser profile sensor. DeepLabv3 model with a ResNet-50 backbone pre-trained on the COCO segmentation dataset was used for the semantic segmentation. The melt pool width has been identified using this approach post the deposition process. However, inclusion of an open-area filter in the post processing means that only the largest continuous segment identified from the training image dataset is retained for identification and classification. This could potentially lead to possible misidentification of the melt pool depending on the presence of artefacts within the images constituting an object larger than the melt pool itself.

In the area of process monitoring, spatter tracking as well as melt pool detection was implemented using 4 different deep learning algorithms with a tracking algorithm built on top to detect the variations in deposition quality [7]. Each spatter particle was assigned a unique ID and tracked temporally using the DeepSORT algorithm. The melt pool stability was analysed using the variation of width, shape and centroid location over time calculated using the implemented segmentation masks. Spatter behaviour was monitored through spatter particle size distribution, ejection angle and speed. The relationship between the melt pool characteristics and the spatter particle behaviour was established based on the observed dynamics. Of all the tested algorithms YOLOv5 was established to be the best suited due to its faster processing time and acceptable accuracy.

In this context, rather than deploying an end-to-end black-box deep learning model directly to raw video frames, this paper proposes and evaluates a progressive, multi-architecture computer vision framework. This multi-tiered evaluation systematically characterizes the trade-offs between computational efficiency, model interpretability, and accuracy, justifying increased model complexity only when simpler baselines demonstrate clear performance deficiencies. Heuristic based methods with rule based feature extraction using ML were implemented in this study. Feature classification

using classical ML pipeline using the selected methods has been implemented. CNN based deep learning models like MobileV2 were then implemented for automated feature learning instead of manual heuristic feature labelling and to perform better on image data. MobileNetV2 was selected due to it being light weight and as it works well on small data and faster training. EfficientNet Temporal modelling - CNN + LSTM - this is used because CNN is being used to extract spacial features and LSTM for learning temporal relationships and to identify defects that are identifiable across multiple frame sequence instead of a single frame.

Finally CNN + LSTM + MCFE was implemented for multiscale feature extraction combining features at different levels (frames) and to capture local fine details and global patterns across multiple frames. All the above constitute the novelty of this work which improves robustness and generalisation of defect identification in WAAM.

The proposed framework begins with raw video frame extraction, followed by a series of preprocessing steps including overlay removal, heuristic background filtering, and weld trail localization. For the classification task, four distinct model families are investigated:

1. **Modular Two-Stage Classical Pipeline:** This approach employs a frozen convolutional neural network (CNN) as a feature extractor, whose outputs are subsequently fed into conventional machine learning classifiers.
2. **End-to-End Deep Learning Models:** Lightweight architectures, including EfficientNet-B0 and MobileNetV2, are utilized to address the limitations associated with fixed feature embeddings and enable fully trainable representations.
3. **Hybrid CNN–LSTM Architecture:** This model combines spatial feature extraction with temporal sequence modeling to capture the onset and progression of weld defects over time.
4. **MCFE-Net (Proposed):** A novel architecture incorporating learnable and differentiable preprocessing modules directly within the end-to-end optimization pipeline, enabling joint optimization of feature extraction and preprocessing stages.

2 Experimentation and Data Collection

Figure. 1 shows the experimental setup used for the wire arc additive manufacturing (WAAM) trials. The deposition system was integrated into an industrial robotic arm to accurately manipulate the welding torch along pre-programmed paths above the build surface. The energy required for material deposition is supplied by a Fronius TPS 400i welding unit, which operates under Cold Metal Transfer (CMT) process modes in a 2-step control configuration. The feedstock material used for deposition is ER307 stainless steel wire, while the substrate material is mild steel. A controlled shielding environment is maintained using a gas mixture composed of 93% argon, 5% carbon dioxide, and 2% oxygen, ensuring arc stability and protection of the molten material from atmospheric contamination. WAAM process operating conditions, such as welding current, arc voltage, wire feed rate, and travel speed, are controlled through the integrated process control system. Table 1 shows the combination of WAAM process parameters used to build the dataset.

A high dynamic range (HDR) monochrome camera was pointed towards the deposition zone to monitor the material deposition process at 30 frames per second (fps). The frame size was set to 1280 x 1024 pixels with a pixel size of 0.0068 mm. This camera system was used to record the critical process phenomena such as the arc shape and morphology, melt pool temporal evolution and shape, spatter formation, droplet transfer, bead solidification and growth. Individual frames from the recorded videos were used to evaluate the process behaviour and deposition consistency. Figure. 1 shows a representative image captured HDR images captured demonstrating the various process phenomena under selected deposition conditions. The process parameter combinations (current, voltage) in the experimental matrix were chosen to cover both stable and unstable bead deposition. In some cases, degraded surface quality was observed due to formation of impurities as well as contaminations due to inadequate shielding. The captured images also included intermittent material transfer yielding non-uniform bead geometry and deposition discontinuities. In some cases, spatter particles due to excessive molten material expulsion were noticed due to unstable arc conditions.

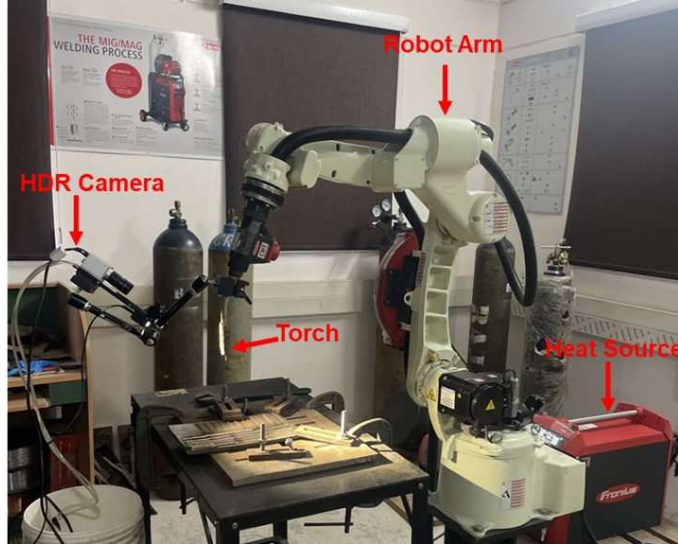


Fig. 1 Experimental Setup

3 Dataset and Filtering

The primary experimental dataset consists of 11 high-resolution video samples (totaling approximately 6,842 frames) captured using a Xiris Automation welding camera. The welding runs were performed under controlled laboratory directed energy deposition conditions, utilizing fixed current, voltage, and torch travel speed settings to isolate visual indicators of weld quality. The camera is equipped with narrow-band spectral filtering centered on the welding arc wavelength. This physical filter suppresses

Table 1 WAAM Process Parameters

Exp. No.	Current (A)	Voltage (V)	Wire Feed Speed (m/min)	Tip Speed (m/min)
1	154	20.7	5.3	55
2	153	20.5	5.3	70
3	129	19.7	4.3	80
4	194	22.7	7.1	80
5	218	25.3	8.0	60
6	139	19.9	4.7	90
7	78	19.1	2.6	20
8	81	18.5	2.6	40
9	80	18.4	2.6	50
10	67	18.6	2.2	50
11	70	18.6	2.2	70

the high-intensity plasma glow, allowing the sensor to capture the underlying boundary of the molten weld pool and the cooling solidifying bead. All frames are extracted as 8-bit grayscale images. The dataset exhibits a severe class imbalance typical of industrial manufacturing background and stable deposition frames constitute the vast majority of samples, while anomalous weld fusion and spatter events are sparsely represented. This distribution directly motivates the resampling and balanced evaluation strategies discussed in Section 4.6.

3.1 Frame Annotation

The video stream is segmented into individual frames, each manually annotated into one of three operational categories: *Background* (idle periods and pre-arc/post-arc phases), *Good* (stable bead deposition with uniform bead width), and *Bad* (representing anomalous deposition states, where beaded and spatter states are merged into a single defect class) as illustrated in Figure 2. From a control systems perspective, merging these defect modes into a single "Bad" category is highly practical: any occurrence of bead height irregularity (beading) or excessive particle ejection (spatter) indicates process instability that requires immediate feed rate or travel speed adjustments. Annotation was conducted using a custom web-based interface built with FastAPI and HTML (Figure 3). The interface supports keyboard-accelerated labeling, minimizing annotation latency and ensuring label consistency.

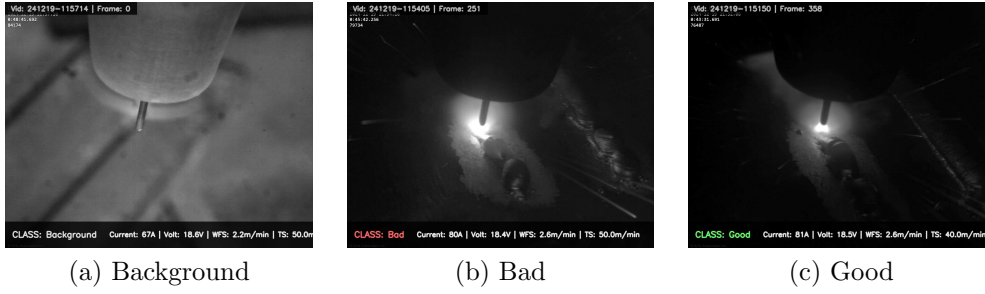


Fig. 2 Annotated frame classes.

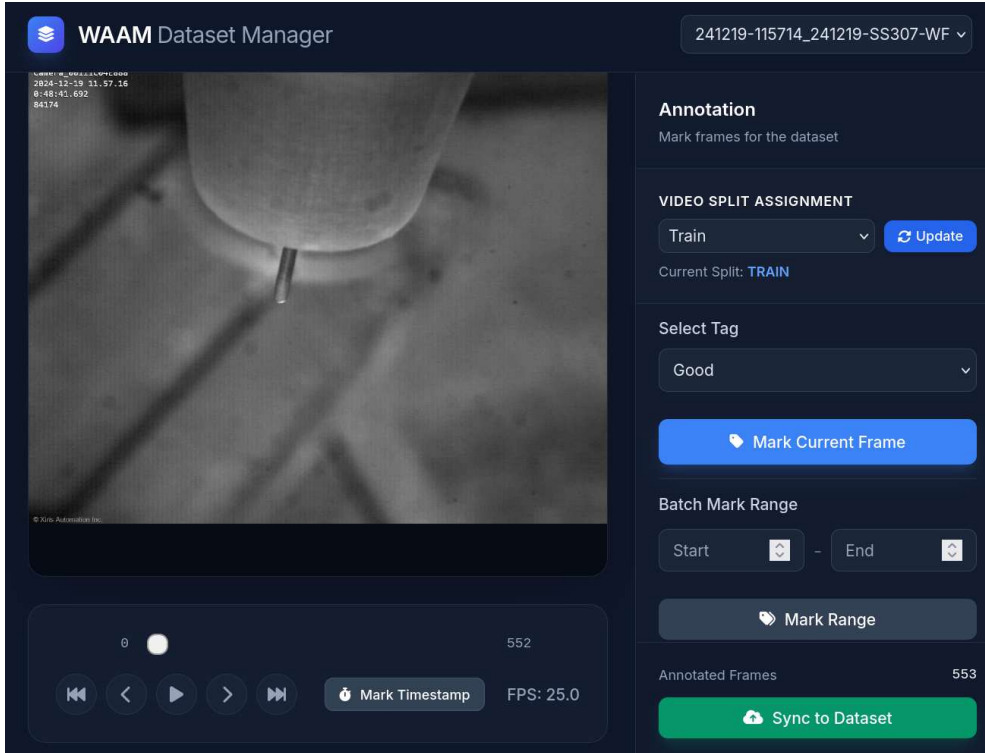


Fig. 3 Custom FastAPI annotation interface for manual frame categorization, supporting keyboard shortcuts across the defined operational states.

3.2 Test-Train Split and Sequential Integrity

A critical challenge in classifying time-series video frames is the high temporal correlation between consecutive frames (captured at 30 frames per second). If frames from the same welding pass are randomly shuffled into training and test splits, the model will see near-duplicate frames in both sets. This "Video Time-propagation" pattern leads to severe data leakage, causing the model to artificially achieve near-100% accuracy by

memorizing specific frame details, while failing to generalize to unseen runs. To prevent this, we partition the dataset at the *video level*: specific video runs (Video IDs) are designated as held-out test and validation sets. This split enforces strict sequential integrity and measures the models’ ability to generalize to entirely unseen physical welding passes under differing substrate thermal histories and setup conditions.

4 Classical Machine Learning

4.1 Motivation for the Three-Step Modular Pipeline

The three-step modular pipeline decouples data pre-filtering, spatial feature extraction, and classification. This separation offers three key practical advantages:

1. **Decoupled Feature Learning:** By leveraging a deep CNN backbone (e.g., MobileNetV2) pre-trained on ImageNet [8], we transfer strong low-level visual priors (such as edge, texture, and boundary detection) directly to the welding domain. This bypasses the need for large, domain-specific labeled datasets to train early convolutional layers.
2. **Computational Efficiency:** The high-dimensional feature embeddings are extracted once and stored in memory. Downstream classical classifiers can then be trained and evaluated in seconds, enabling rapid hyperparameter exploration without the overhead of repeating backpropagation through deep convolutional blocks.
3. **Model Interpretability:** Classifiers such as Logistic Regression and Decision Trees offer explicit mathematical formulations of decision boundaries. This allows operators to inspect feature importances and understand which visual representations correlate with weld defects, establishing a transparent baseline before moving to end-to-end models.

However, training a deep neural network end-to-end on a small, imbalanced dataset of approximately 7,000 frames risks severe overfitting, particularly for the minority classes. To address this, resampling was performed in order to increase the performance on the minority defect class (Bad) by mitigating the severe class imbalance in the training set. We evaluated multiple class-balancing techniques, including Random Oversampling (ROS), SMOTE, and image-level augmentations (flipping and rotation) to broaden the representation of defect modes in the embedding space.

We chose to resample the defect frames to 40% of the total stable deposition frames. This ratio was selected to achieve a balanced performance: it avoids the overfitting associated with higher resampling ratios while preventing the low recall that occurs when the count of minority defect frames in the training sample is too low. We tried different values for the resampling percentage and evaluated the performance of a baseline MobileNetV2 model with Logistic Regression. We observed a performance improvement after resampling (Unbalanced Defect F1-Score: 0.308 vs. Balanced Defect F1-Score: 0.350). The class distributions before and after resampling are shown in Figure 4 and Figure 5, respectively.



Fig. 4 Class distribution before resampling, showing severe class imbalance for the minority defect class.



Fig. 5 Class distribution after resampling (minority defects oversampled to 40% of stable frames).

4.2 Heuristic Background Filtering

The raw video sequence contains substantial idle periods before arc ignition and after extinguishment (e.g., torch positioning, pre-purge gas flow, and post-weld cooling) that do not represent active welding deposition. Including these frames in training would bias the downstream classifiers toward predicting background, masking critical weld defect signatures. We filter them using a dual-metric heuristic combining mean pixel intensity and per-frame variance across the video sequence.

The welding arc is an extreme, localized high-brightness event. Mean intensity alone is insufficient to distinguish this event from ambient light drift; pairing it with frame variance resolves the ambiguity. As shown in Figure 6, idle segments exhibit a low, stable mean intensity with minimal variance, while active deposition zones show a characteristic sharp peak in both metrics—driven by arc ignition, dynamic spatter ejection, and pool ripple. Threshold selection on these co-occurring metrics isolates the active deposition window, removing idle frames before model training.

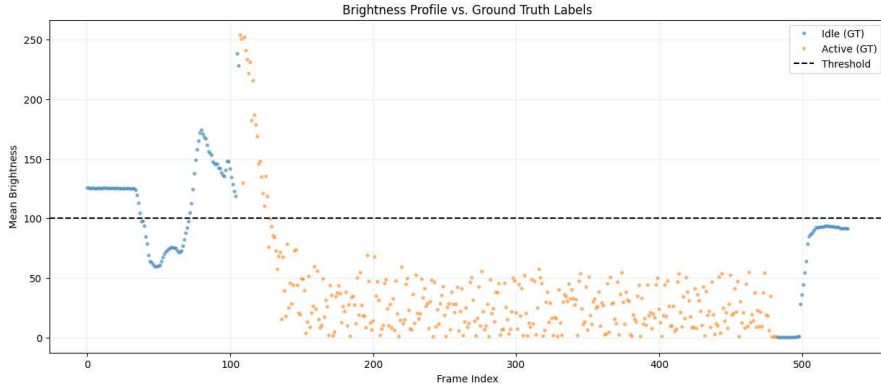


Fig. 6 Per-frame mean brightness and variance across the video sequence. Sharp co-occurring peaks in both metrics reliably identify active deposition windows.

4.2.1 Initial Heuristic Segmentation

The initial bounding-body identification relied on traditional image processing techniques: adaptive thresholding, Canny edge detection, and contour approximation. While computationally inexpensive, this approach is severely confounded by fluctuating arc glare and flying spatter particles, which share the same high-intensity pixel signature as the weld pool boundary. As a result, the contour detector is highly unstable, leading to a classification accuracy of approximately **70%** and a poor recall of **0.67** for the Active class (Table 2). This low recall means a significant portion of active deposition frames are incorrectly filtered or misaligned, degrading downstream feature quality.

Class	Precision	Recall	F1-score	Support
Idle	0.67	0.97	0.79	2753
Active	0.97	0.67	0.79	4009
Accuracy			0.79	6762
Macro Avg	0.82	0.82	0.79	6762
Weighted Avg	0.85	0.79	0.79	6762

Table 2 Classification Report: Before Single Segment Block Summation

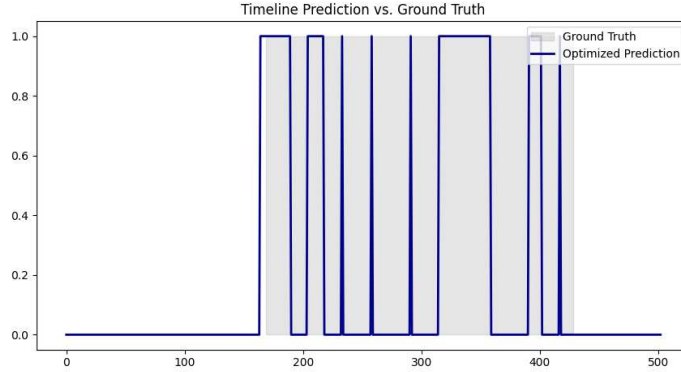


Fig. 7 Bounding body identification comparison: the 70% heuristic (left) vs. 90% segmentation (right).

4.2.2 Continuous Segmentation Block

To resolve the limitations of threshold-based contour isolation, we implement a Continuous Segmentation Block. This method exploits the temporal and spatial persistence of the weld pool and cooling trail. Unlike spatter and arc flares, which are transient and disappear across consecutive frames, the molten pool and solidified bead are spatially contiguous and temporally persistent. The block tracks this spatially persistent

region by generating a dense segmentation composite over consecutive frames. This temporal tracking allows the algorithm to semantically separate the actual weld pool boundary from transient spatter artifacts and arc reflections.

As shown in Table 3 and Figure 8, the continuous segmentation block improves identification accuracy to **84%** and raises the Active class recall to **1.00**. This means all active deposition frames are correctly identified and centered, providing clean, geometrically aligned input crops for all downstream classification models.

Class	Precision	Recall	F1-score	Support
Idle	1.00	0.67	0.81	243
Active	0.77	1.00	0.87	260
Accuracy			0.84	503
Macro Avg	0.88	0.84	0.84	503
Weighted Avg	0.88	0.84	0.84	503

Table 3 Segment Logic Performance Report

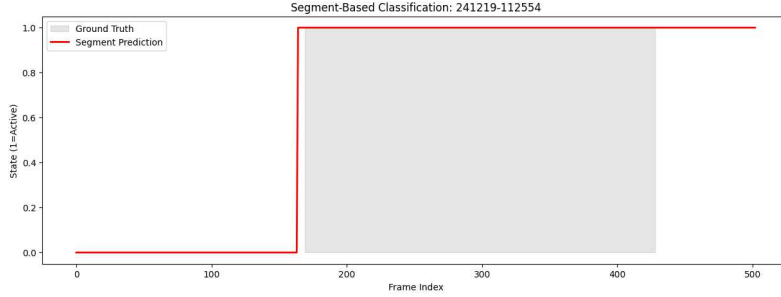


Fig. 8 Confusion matrix and timeline prediction vs. ground truth following continuous segmentation block processing.

4.2.3 Limitations of Segmentation

While the continuous segmentation block achieves a perfect active recall, it operates under two key physical assumptions: (1) active deposition is characterized by a single monotonic ignition-to-extinguishment step (2) high-brightness events are always associated with active welding. In multi-pass welding, long inter-pass cooling pauses may violate the monotonicity assumption, causing idle passes to be incorrectly retained. Furthermore, pre-ignition arc flashes can trigger the active threshold prematurely. Future work should investigate multi-modal segmentation models that combine optical frame tracking with real-time current and voltage measurements to build a more robust tracking gate.

4.3 Preprocessing

The primary challenge in preprocessing WAAM imagery is the extreme optical dynamic range of the arc plasma. The plasma core overexposes the center of the frame, while the cooling solidification trail is underexposed. Effective preprocessing must suppress the blinding arc glare, enhance the visibility of the solidification ripples, and provide the CNN backbones with highly discriminative spatial signals. Figure 9 illustrates the visual output of each preprocessing step.

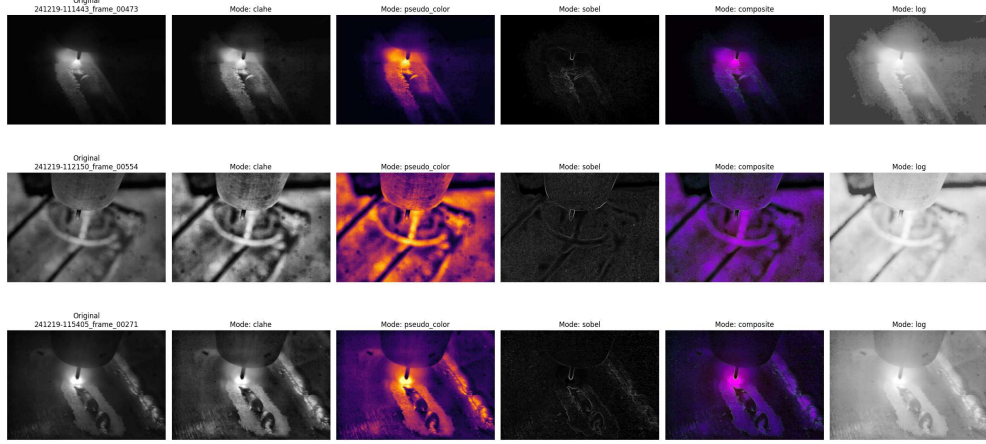


Fig. 9 Preprocessing pipeline: raw grayscale frame (leftmost) through CLAHE, Sobel edge map, logarithmic transform, and temporal difference. Each transform targets a specific perceptual deficiency of the raw frame.

4.3.1 CLAHE

Contrast Limited Adaptive Histogram Equalization [1] equalizes the frame contrast locally within tile regions. This local equalization prevents the high-intensity arc plasma core from dominating the global intensity histogram, suppressing glare at the frame center while revealing the low-contrast solidifying bead boundary in the cooling trail—a region critical for distinguishing Good and Bad fusion states.

4.3.2 Multi-Channel Composite

Because input frames are grayscale, passing them directly to ImageNet-pretrained CNN backbones would waste two of the three expected input channels. We construct a 3-channel composite: Channel R represents the CLAHE-enhanced frame (structural contrast), Channel G represents the Sobel edge map (geometric texture of bead ripples and spatter boundaries), and Channel B represents the raw grayscale frame (ground-truth intensity). This composite assigns distinct physical features to each color channel, triggering the early convolutional filters of the ImageNet backbone to separately process contrast, edge, and intensity representations [9].

4.3.3 Temporal Differencing

Weld defects (such as spatter and bead instability) are highly dynamic events. A single frame may show normal bead geometry, but the temporal difference $|I_t - I_{t-1}|$ highlights moving spatter particles and localized pool oscillations. Stacking this motion residual as an input channel provides the static CNN with a proxy for temporal change, allowing it to capture dynamics without requiring a recurrent network.

4.3.4 Logarithmic Transform

The dynamic range of the welding arc can exceed what CLAHE’s local tiles can compensate. A logarithmic transform $c \cdot \log(1 + I)$ compresses this global intensity range, preventing saturation at the arc core without generating the local halo artifacts that CLAHE occasionally produces at high-contrast boundaries.

4.4 CNN Backbones for Feature Extraction

We evaluate three deep CNN backbones as feature extractors, balancing parameter count, representation capacity, and inference speed. All backbones are initialized with ImageNet pretrained weights, and their final classification heads are removed. This produces a fixed-dimensional embedding vector for each input frame, which is passed to downstream classical classifiers.

Table 4 CNN Backbones Used for Feature Extraction

Model	Parameters	Training Time	Expected Performance
MobileNetV2	3.4M	Fast	Baseline
EfficientNet-B0	5.3M	Moderate	Balanced
ResNet-50	25.6M	Slow	High capacity

The choice of backbone is deliberate: MobileNetV2 is selected to test whether a lightweight backbone can extract sufficient visual cues for WAAM quality monitoring; EfficientNet-B0 tests compound-scaled features [10] that optimize accuracy and parameter count; and ResNet-50 [11] tests whether additional depth and residual connections improve embedding quality despite the limited dataset (Table 4).

4.5 Classical Classifiers

The extracted feature embeddings are passed to two classical classifiers to perform a binary classification task distinguishing between *Background* (idle state, mapped to 0) and *Active* (active welding, mapped to 1) operational states. This binarization establishes a baseline detector for active deposition. Each classifier represents a distinct hypothesis about the structure of the embedding space:

1. **Logistic Regression** Tests whether the CNN-generated feature embeddings are linearly separable. If a linear classifier suffices, it indicates that the frozen CNN backbone is doing most of the representational work.

2. **Random Forest:** Ensembles multiple decision trees to reduce variance, mitigating overfitting in the high-dimensional feature space.

4.6 Results and Analysis

4.6.1 Top Configurations

Across all 24 evaluated combinations (representing 4 preprocessing techniques $\times$ 3 CNN backbones $\times$ 2 classical classifiers), the top-performing configuration was **CLAHE + EfficientNet-B0 + Random Forest**, which achieved an overall accuracy of 0.997, a weighted F1-score of 0.997, and an Active class recall of 0.995. Table 5 compiles the performance metrics for the key configurations on the held-out test split under their best-performing preprocessing modes, and Figure 10 visualizes the trade-off between the global F1-score and Active class recall.

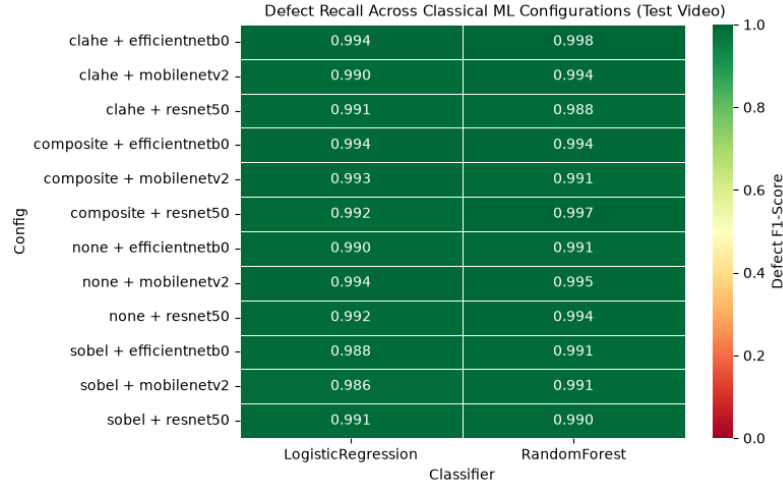


Fig. 10 Overall F1-score vs. Active class recall for all 24 configurations under binary classification.

Backbone	Preprocessing	Classifier	Acc.	F1 (W)	Active Recall
MobileNetV2	None	Rand. Forest	0.995	0.995	0.991
MobileNetV2	None	Logistic Reg.	0.993	0.993	0.988
EfficientNet-B0	CLAHE	Rand. Forest	0.997	0.997	0.995
EfficientNet-B0	CLAHE	Logistic Reg.	0.993	0.993	0.991
ResNet-50	Composite	Rand. Forest	0.996	0.996	0.995
ResNet-50	None	Logistic Reg.	0.991	0.991	0.993

Table 5 Performance comparison of Backbone and Classifier combinations under best preprocessing modes on the test set.

4.6.2 Component Analysis

Averaging the evaluation results across the different configurations isolates the individual contribution of each component:

- **Best Classifier — Random Forest** Random Forest achieved a slightly higher average F1-score of 0.992 compared to Logistic Regression’s 0.990, and a higher Active class recall (0.994 vs. 0.991). Under the binarized classification task (Background vs. Active), both linear and non-linear classifiers generalize extremely well on the test split. The ensemble tree structures of Random Forest capture subtle non-linear variance in the embedding space once the classification task is simplified to active state detection.
- **Best Backbone — EfficientNet-B0:** EfficientNet-B0 achieved the highest mean F1-score (0.991) and mean Active class recall (0.994), outperforming both MobileNetV2 (F1: 0.991, recall: 0.992) and ResNet-50 (F1: 0.991, recall: 0.992). The compound-scaled spatial features of EfficientNet-B0 transfer robustly to binarized WAAM frame representations.
- **Best Preprocessing — Composite:** The Composite preprocessing method produced the highest average weighted F1-score (0.992) and Active class recall (0.994), illustrating the benefit of combining CLAHE, Sobel, and raw grayscale channels. None and CLAHE also performed strongly (average F1 of 0.992 and 0.991 respectively), while Sobel had a slightly lower overall accuracy but high recall.

For optimal generalization on the test split, the combination of **CLAHE preprocessing, EfficientNet-B0, and Random Forest** represents the best classical baseline, achieving an Active class recall of 0.995. The key limitation of this modular architecture is the information bottleneck at the embedding stage: because the CNN backbone is frozen, its feature extraction parameters cannot be adjusted based on classification error, and the preprocessing remains fixed and hand-crafted. Furthermore, this baseline is restricted to binary active state detection. This limitation motivates the transition to end-to-end deep learning for fine-grained 3-class quality monitoring.

5 End-to-End Deep Learning

The two-step architecture demonstrates strong overall accuracy but exposes a fundamental bottleneck: the CNN backbone is frozen and receives no gradient signal from the classification loss. The preprocessing is likewise fixed, determined entirely by hand-crafted heuristics. In end-to-end training, all parameters—from the first convolutional layer to the final classifier head—are jointly optimized against the classification objective, allowing the network to learn representations specifically adapted to the industrial features of WAAM imagery.

5.1 MobileNetV2 End-to-End

As a baseline for end-to-end training, we fine-tune MobileNetV2 [12] by replacing its 1000-class ImageNet head with a 3-class linear classifier and training all parameters jointly. This directly tests whether joint gradient optimization overcomes the

representational bottleneck of the frozen backbone. The training and validation convergence curves are plotted in Figure 11. The final test set confusion matrix is shown in Figure 12, and the detailed test split performance is summarized in Table 6.

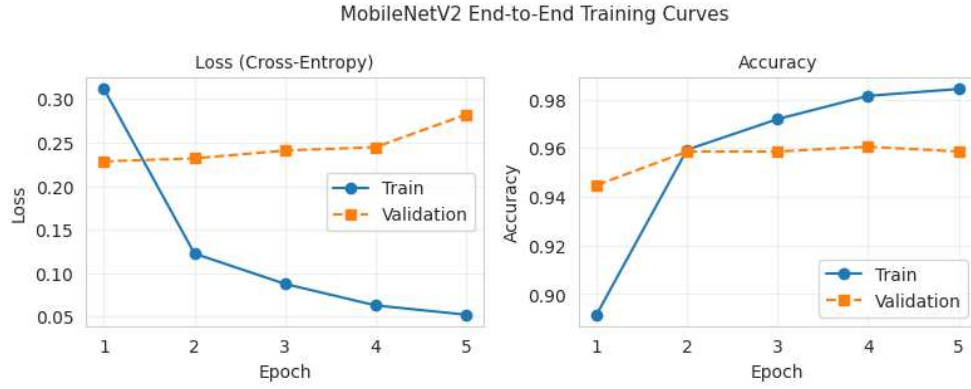


Fig. 11 Training and validation loss and accuracy curves for MobileNetV2 end-to-end fine-tuning, demonstrating rapid convergence and stable generalization.

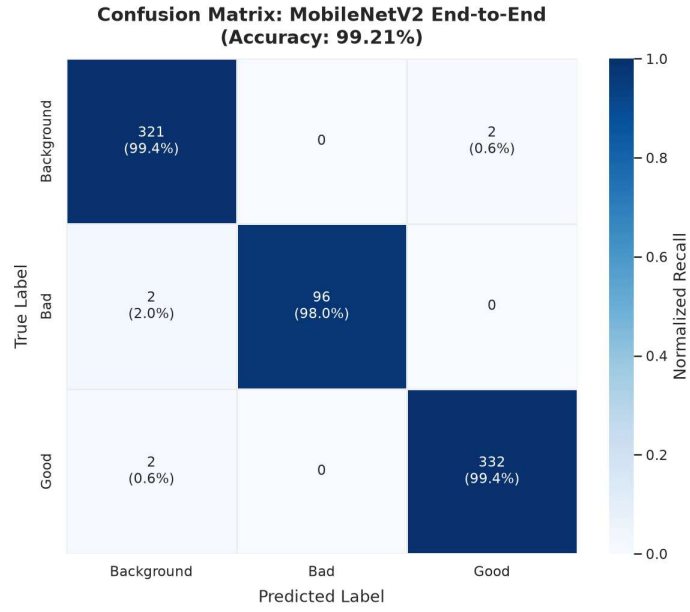


Fig. 12 Confusion matrix (3-class) for MobileNetV2 on test split.

Class	Prec.	Recall	F1	Support
Background	0.99	0.99	0.99	323
Bad	1.00	0.98	0.98	98
Good	0.99	0.99	0.99	334
Accuracy			0.99	755
Macro Avg	0.99	0.99	0.98	755
Weighted Avg	0.99	0.98	0.98	755

Table 6 MobileNetV2 — Test Video Evaluation

End-to-end fine-tuning of MobileNetV2 immediately overcomes the frozen-backbone bottleneck. All classes achieve near-perfect metrics, with the Bad class reaching an F1-score of 0.99 and a recall of 0.98. This confirms that the joint optimization allows the compact network to learn domain-specific spatial filters that isolate the weld pool boundaries and spatter particles. The inverted residuals and linear bottlenecks restrict the network capacity, which acts as a strong regularizer that prevents overfitting to background noise despite the small training dataset.

5.2 EfficientNet-B0 End-to-End

EfficientNet-B0 [10] is evaluated end-to-end to test the efficacy of compound scaling—which scales depth, width, and resolution under a fixed parameter budget—in comparison to MobileNetV2. The model was trained with the Adam optimizer [13] (learning rate = 10^{-4}), cross-entropy loss, and standard augmentations. The training curves are shown in Figure 13, the test split confusion matrix is shown in Figure 14, and the detailed test split performance is summarized in Table 7.

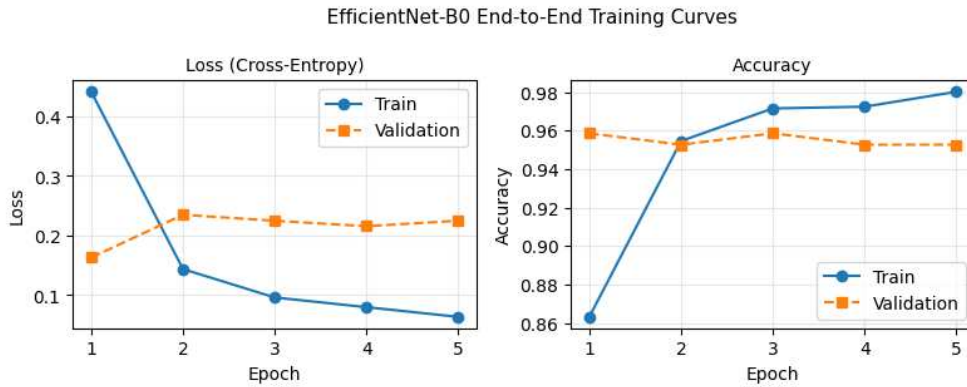


Fig. 13 Training and validation loss and accuracy curves for EfficientNet-B0 end-to-end fine-tuning.

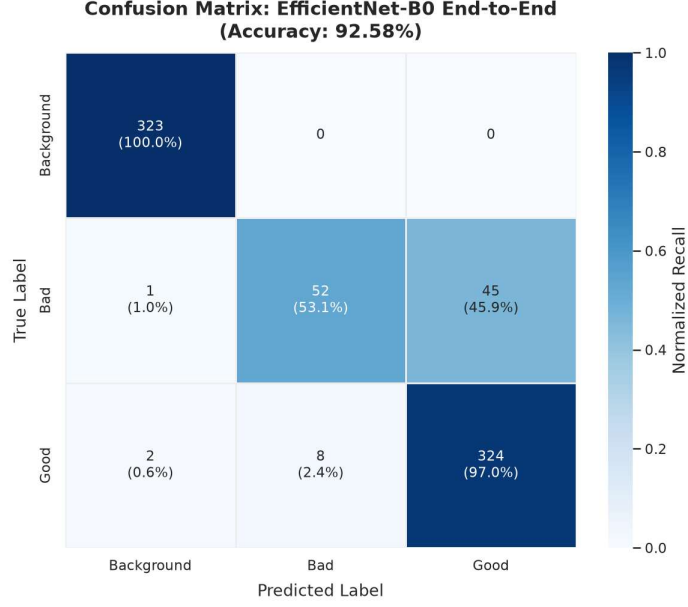


Fig. 14 Confusion matrix for EfficientNet-B0 on test split.

Class	Prec.	Recall	F1	Support
Background	0.99	1.00	1.00	323
Bad	0.87	0.53	0.66	98
Good	0.88	0.97	0.92	334
Accuracy			0.93	755
Macro Avg	0.91	0.83	0.86	755
Weighted Avg	0.93	0.93	0.92	755

Table 7 EfficientNet-B0 — Test Video Evaluation

EfficientNet-B0 achieves a weighted-average accuracy of 93% on the test split, but suffers from a significant drop in defect recall (0.53) compared to MobileNetV2. Because EfficientNet-B0 has a larger parameter space and utilizes compound scaling, it has a higher representational capacity. On a relatively small dataset, this larger capacity makes the model prone to getting trapped in local minima during backpropagation, leading it to overfit to the dominant classes (Background and Good) while failing to generalize to the minority defect class (Bad). This result underscores that larger deep learning architectures do not necessarily translate to better performance under data constraints and high class imbalance.

6 Recurrent Convolutional Architecture

EfficientNet-B0’s per-frame classification ignores the temporal context that distinguishes a developing defect event from a single transient arc fluctuation. From a process physics standpoint, WAAM defects are thermodynamic transitions with measurable onset dynamics: a defect event begins as a subtle change in pool width that progresses over multiple frames before becoming visually definitive. A classifier that treats each frame independently cannot model this progression and is susceptible to both false positives and false negatives. Sequential sample frames illustrating the temporal nature of WAAM defects are shown in Figure 15.

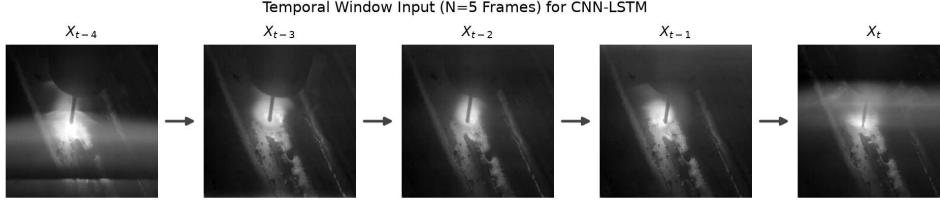


Fig. 15 Sequential video frame samples illustrating the temporal nature of WAAM deposition.

6.1 CNN-LSTM Architecture

We implement a hybrid CNN-LSTM architecture in which a shared EfficientNet-B0 backbone processes each frame in a temporal window independently, and the resulting embedding sequence is consumed by an LSTM [14] layer that models temporal dependencies. While alternative models such as Convolutional LSTM (ConvLSTM) [15] integrate spatial and temporal features within recurrent transitions, our decoupled design enables the reuse of robust, pretrained spatial feature extractors.

6.1.1 Spatial Feature Extraction

EfficientNet-B0 serves as the shared spatial encoder. Its weights are shared across all frames in the sequence, ensuring that the spatial feature extractor is consistent across time steps and that the model does not overfit to positional ordering of frames within the window.

6.1.2 Temporal Sequence Modeling

The LSTM [14] receives the sequence of embedding vectors $(V_{t-N}, \dots, V_t)$ and maintains a hidden state h_t that accumulates temporal context. The forget, input, and output gates of the LSTM allow it to selectively retain information about gradual pool deformation while discarding transient single-frame anomalies (spurious bright pixels from lens flare). The final hidden state h_t encodes the temporal trajectory of the welding process over the entire window.

6.1.3 Classification

A fully connected layer maps h_t to a probability distribution over the 3 operational classes. The end-to-end gradient signal during training flows from the classification loss backward through the FC layer, LSTM, and into the shared CNN backbone, allowing the spatial encoder to develop features that are temporally informative rather than optimized solely for single-frame classification.

Forward Pass:

1. Input: temporal sequence $(X_{t-N}, \dots, X_t) \in \mathbb{R}^{N \times H \times W \times 3}$.
2. Distributed feature extraction: CNN backbone f_θ maps each frame to $V_i = f_\theta(X_i) \in \mathbb{R}^d$.
3. Recurrence: LSTM iterates $(V_{t-N}, \dots, V_t) \rightarrow h_t$.
4. Classification: $\hat{y} = \text{softmax}(Wh_t + b)$.

The LSTM’s memory mechanism provides intrinsic noise robustness: a transient arc flash that produces an anomalous embedding at a single time step is dampened by the gate mechanisms and does not propagate to dominate the final hidden state—substantially reducing false positive defect detections.

A key limitation of the CNN-LSTM remains the fixed preprocessing applied before the input sequence is constructed. Regardless of welding conditions, the network receives frames processed by the same hand-crafted transforms. This motivates the following section, which proposes making the preprocessing itself a learnable component of the architecture.

6.1.4 Results

The final confusion matrix of the CNN-LSTM model is displayed in Figure 16. The detailed classification report on the test split is provided in Table 8. The CNN-LSTM achieves 94% accuracy and raises Bad class F1 to 0.74—the strongest single-class defect score across all models. The LSTM hidden state provides exactly the temporal context needed to distinguish a developing Bad-deposition sequence (sustained arc irregularity across 5 frames) from isolated noise. False positive defect detections are substantially reduced compared to the per-frame EfficientNet model, as the gate mechanisms suppress single-frame anomalies. The remaining limitation is fixed hand-crafted preprocessing, which motivates MCFE-Net.

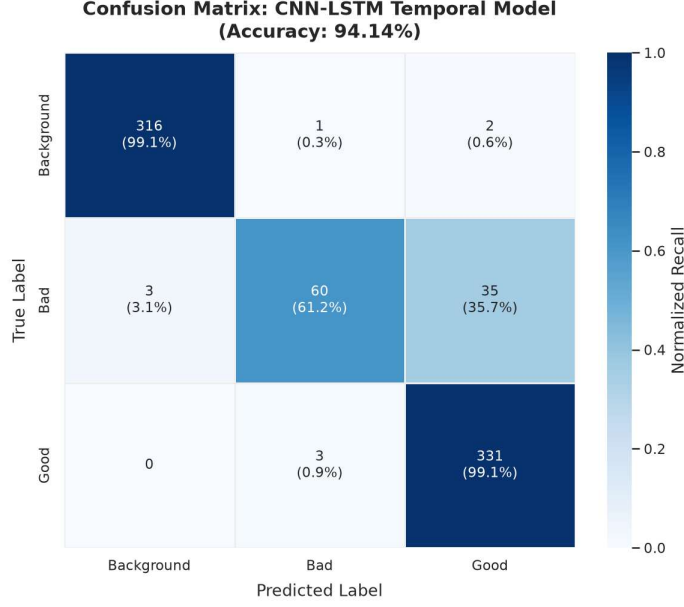


Fig. 16 Confusion matrix for CNN-LSTM on test split.

Class	Prec.	Recall	F1	Support
Background	0.99	0.99	0.99	319
Bad	0.94	0.61	0.74	98
Good	0.90	0.99	0.94	334
Accuracy			0.94	751
Macro Avg	0.94	0.86	0.89	751
Weighted Avg	0.94	0.94	0.94	751

Table 8 CNN-LSTM — Test Video Evaluation

7 MCFE - Multi-Channel Feature Engineering Network

The experimental results from Section 4.6 demonstrate two key points: (1) different fixed preprocessing techniques (such as Sobel, CLAHE, or composite methods) have a substantial and consistent impact on quality classification, as shown by the variance in Table 5; and (2) the best-performing preprocessing method varies depending on the choice of downstream classifier, indicating that the best feature representation cannot be determined a priori. The natural resolution is to make the preprocessing stage

differentiable and jointly optimize it with the classification objective. This allows the network to automatically discover the optimal preprocessing strategy that minimizes classification loss on the specific WAAM dataset.

7.1 MCFE Architecture

MCFE-Net (Multi-Channel Feature Engineering Network) replaces the fixed preprocessing pipeline with two small, learnable convolutional modules: the Contrast Enhancement Module (CEM) and the Edge Saliency Module (ESM). These modules dynamically synthesize the second and third input channels from the raw grayscale frame. They are initialized with weights that mimic CLAHE and Sobel filters, respectively, but are updated via backpropagation. Figure 17 illustrates the channel construction flow.

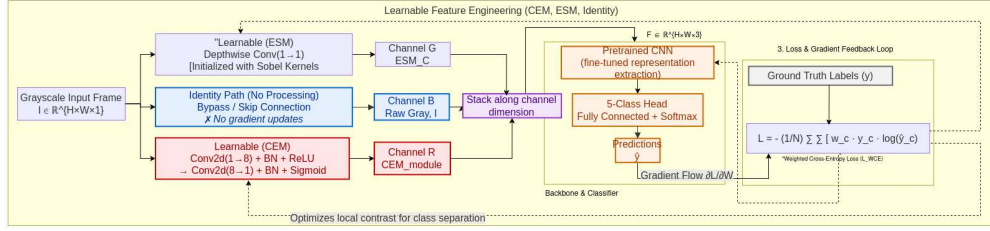


Fig. 17 MCFE-Net channel construction. The raw grayscale frame is processed by the learnable CEM and ESM modules to produce channels R and G respectively; channel B is the raw frame. The three channels are stacked and passed to the CNN backbone.

Contrast Enhancement Module (CEM) A two-layer convolutional network that approximates local contrast normalization:

$$\hat{C} = \sigma(\text{BN}(\text{Conv}_{3 \times 3}^{8 \rightarrow 1}(\text{ReLU}(\text{BN}(\text{Conv}_{3 \times 3}^{1 \rightarrow 8}(I)))))) \quad (1)$$

The sigmoid activation constrains the output to $[0, 1]$, matching the intensity range expected by ImageNet-pretrained weights.

Edge Saliency Module (ESM). A single depthwise convolutional layer initialized with a Sobel-x kernel:

$$\hat{E} = \text{ReLU}(\text{BN}(\text{Conv}_{3 \times 3}^{1 \rightarrow 1}(I; W_0 = W_{\text{Sobel}}))) / \max \quad (2)$$

The Sobel initialization provides a known-good prior; backpropagation allows the kernel weights to adapt to highlight the specific geometric features (such as cooling ripples and weld pool boundaries) most discriminative for WAAM classification.

Full forward pass:

$$F = [\hat{C}(I) \parallel \hat{E}(I) \parallel I] \in \mathbb{R}^{H \times W \times 3} \quad (3)$$

$$\hat{y} = \text{CNN-Backbone}(F) \quad (4)$$

7.2 Training Protocol

To train MCFE-Net, we define two learning rate groups: a higher rate (5×10^{-4}) for CEM and ESM to allow rapid adaptation from their initialization priors, and a standard fine-tuning rate (10^{-4}) for the CNN backbone. For the first two epochs, the backbone weights are frozen to allow the learnable preprocessing modules to stabilize before joint optimization. Class-weighted cross-entropy is used, with the Bad class up-weighted to handle the class imbalance. The training curves for MCFE-Net (EfficientNet-B0 backbone) are shown in Figure 18. The detailed test split evaluation metrics are compiled in Table 9, Table 10, and Table 11.

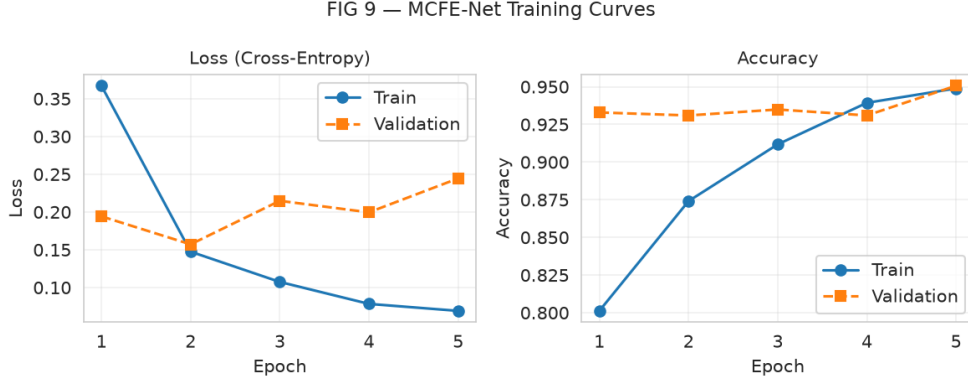


Fig. 18 Training and validation curves for MCFE-Net (EfficientNet backbone) showing the joint optimization of learnable preprocessing modules (CEM, ESM) and classification layers.

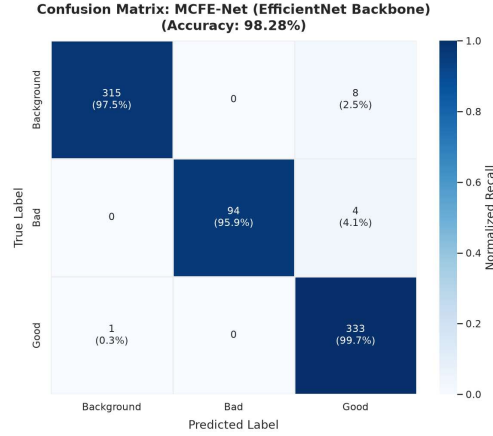


Fig. 19 Confusion matrix (3-class) for MCFE-Net (EfficientNet backbone) on test split

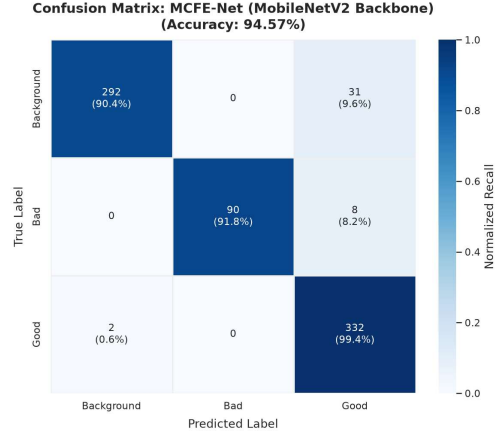


Fig. 20 Confusion matrix (3-class) for MCFE-Net (MobileNetV2 backbone) on test split.

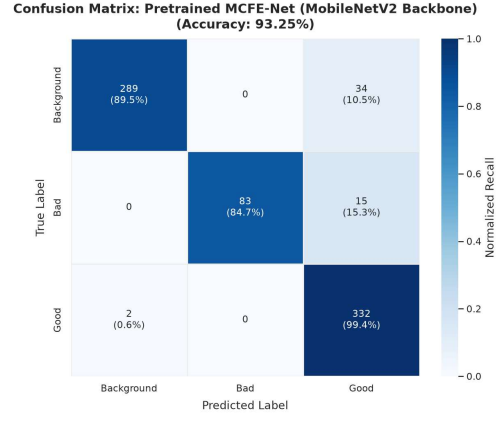


Fig. 21 Confusion matrix (3-class) for Pretrained MCFE-Net (MobileNetV2 backbone) on test split.

Table 9 MCFE-Net(EfficientNetBackbone)—TestVideo Evaluation

Class	Prec.	Recall	F1	Support
Background	1.00	0.98	0.99	323
Bad	1.00	0.96	0.98	98
Good	0.97	1.00	0.98	334
Accuracy			0.98	755
Macro Avg	0.99	0.98	0.98	755
Weighted Avg	0.98	0.98	0.98	755

Table 10 MCFE-Net (MobileNetV2 Backbone)—Test Video Evaluation

Class	Prec.	Recall	F1	Support
Background	0.99	0.90	0.95	323
Bad	1.00	0.92	0.96	98
Good	0.89	0.99	0.94	334
Accuracy			0.95	755
Macro Avg	0.96	0.94	0.95	755
Weighted Avg	0.95	0.95	0.95	755

Table 11 Pretrained MCFE-Net (MobileNetV2 Backbone) — Test Video Evaluation

Class	Prec.	Recall	F1	Support
Background	0.99	0.89	0.94	323
Bad	1.00	0.85	0.92	98
Good	0.87	0.99	0.93	334
Accuracy			0.93	755
Macro Avg	0.95	0.91	0.93	755
Weighted Avg	0.94	0.93	0.93	755

7.3 Interpretability Contribution

A key advantage of MCFE-Net is its physical interpretability. The learned CEM and ESM convolutional kernels can be visualized and compared to OpenCV transforms (Figure 22), revealing which visual features are most informative for defect detection. As shown in Table 9, MCFE-Net (EfficientNet backbone) achieves an accuracy of 98%, with perfect precision (1.00), high recall (0.96) for the Bad class. The MCFE-Net (MobileNetV2 backbone) also performs strongly with 95% accuracy and perfect precision (1.00) for the Bad class (Table 10). The perfect precision is highly beneficial for industrial monitoring, as it eliminates costly false alarms. This confirms that jointly optimizing preprocessing filters with classification layers yields highly discriminative, noise-robust features.

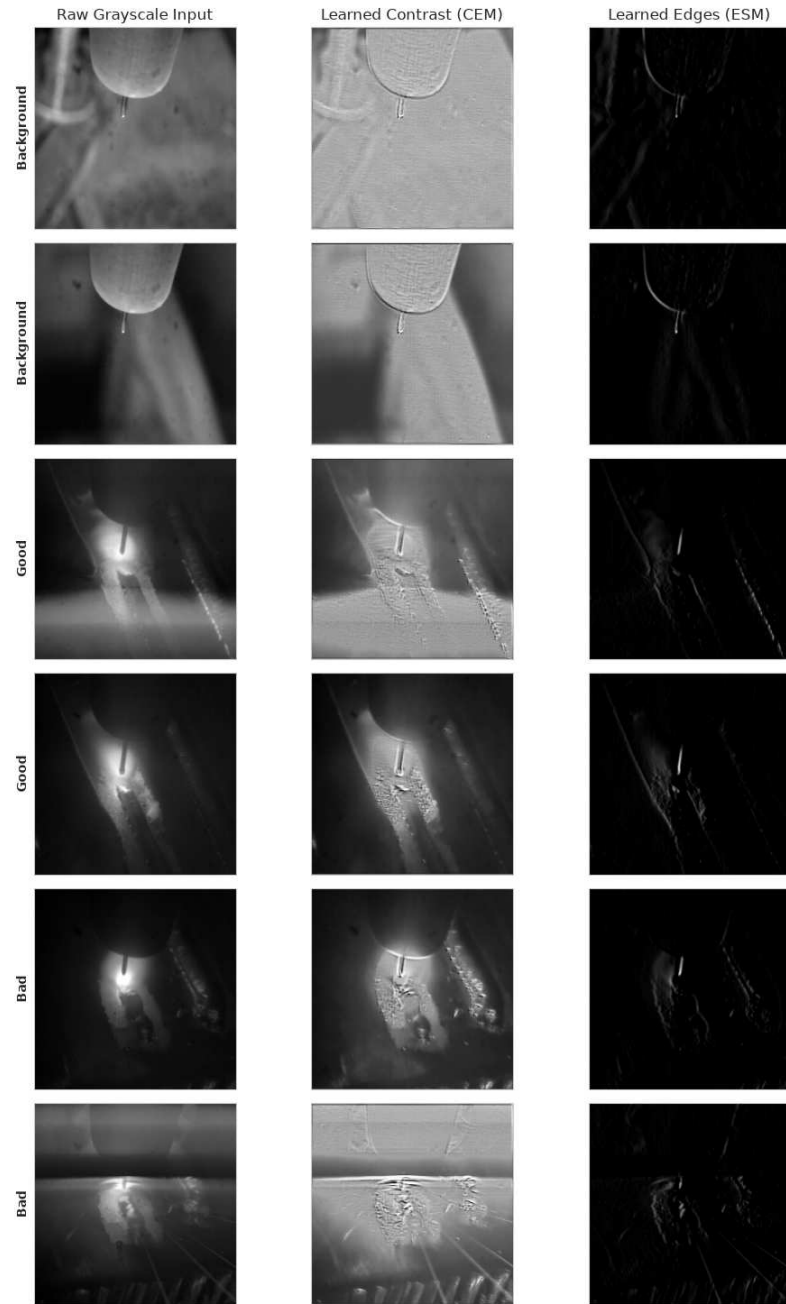


Fig. 22 Learned preprocessing channels (CEM and ESM).

7.4 Parameter Overhead

CEM adds approximately 160 parameters, and ESM adds 10 parameters. The total overhead relative to the backbones (MobileNetV2 or EfficientNet-B0) is negligible ($< 0.004\%$). Consequently, MCFE-Net retains the backbones' inference speed profiles while gaining adaptive preprocessing.

7.5 MCFE-CNN-LSTM (Proposed)

To fully leverage both learnable spatial features and temporal sequence dynamics, we propose the hybrid **MCFE-CNN-LSTM** architecture. In this design, the learnable CEM and ESM modules are prepended to a shared spatial backbone (MobileNetV2) to dynamically reconstruct the input channels (Figure 17), and the sequence of spatial embeddings extracted across a sliding 5-frame window is sequentially modeled by an LSTM layer, figure 23 presents the full architecture block diagram.

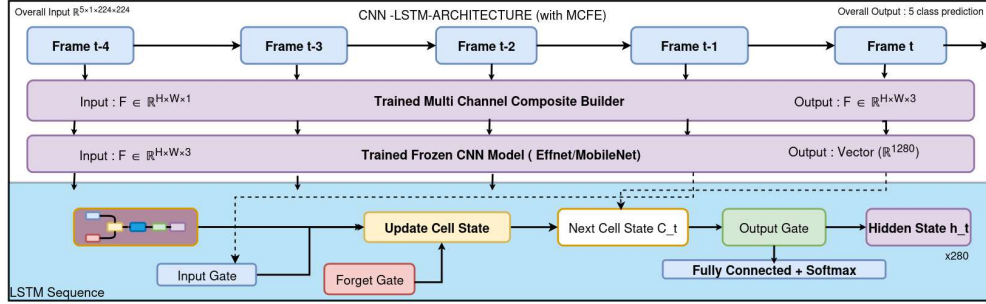


Fig. 23 Architecture block diagram of the CNN-LSTM model, showing frame-by-frame feature extraction followed by recurrent sequence modeling.

The classification report for the MCFE-CNN-LSTM model on the held-out test video sequence is presented in Table 12.

Table 12 MCFE-CNN-LSTM (Proposed) — Test Video Evaluation

Class	Prec.	Recall	F1	Support
Background	1.00	0.97	0.99	319
Bad	1.00	0.98	0.99	98
Good	0.97	1.00	0.99	334
Accuracy			0.99	751
Macro Avg	0.99	0.98	0.99	751
Weighted Avg	0.99	0.99	0.99	751

The proposed MCFE-CNN-LSTM architecture integrates learnable preprocessing, spatial feature extraction, and temporal sequence modeling, achieving an accuracy and

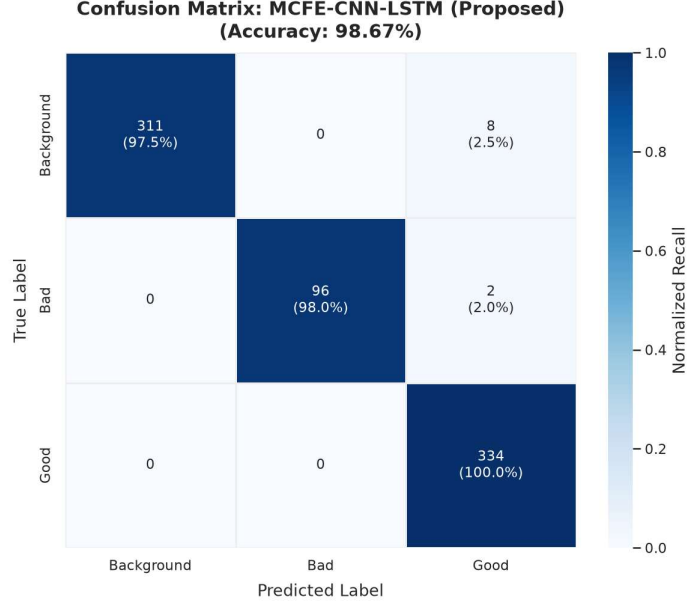


Fig. 24 Confusion matrix (3-class) for the proposed MCFE-CNN-LSTM on the test video frames.

weighted F1 of 0.99. The Bad class F1-score reaches 0.99 with a recall of 0.98, demonstrating that combining learnable preprocessing with temporal recurrent modeling resolves the limitations of per-frame stateless architectures. It eliminates false positives caused by transient arc flares while retaining high sensitivity to weld pool geometry drift. Furthermore, it retains full interpretability, allowing engineers to visualize the learned CEM and ESM filters.

7.6 Overall Architectural Comparison

Table 13 compiles the key metrics across all eight evaluated configurations, from the two-stage classical baseline to the proposed MCFE-CNN-LSTM hybrid.

Table 13 Overall Comparative Evaluation on Test Video Split — All Architectures

#	Model / Architecture	Defect Recall	Overall F1	Methodology
1	Best Classical (CLAHE + EffNet-B0 + RF)	0.995	0.997	Two-Stage CNN Embedding + Classical ML (Binary)
2	End-to-End MobileNetV2	0.980	0.991	End-to-End Fine-Tuning
3	End-to-End EfficientNet-B0	0.531	0.922	End-to-End Fine-Tuning
4	CNN-LSTM (EfficientNet Backbone)	0.612	0.943	Temporal Sliding-Window Sequence
5	MCFE-Net (EfficientNet Backbone)	0.959	0.983	Learnable Preprocessing + Fine-Tuning
6	MCFE-Net (MobileNetV2 Backbone)	0.918	0.946	Learnable Preprocessing + Fine-Tuning
7	MCFE-Net (Pretrained MobileNetV2 Backbone)	0.847	0.933	Learnable Preprocessing + Fine-Tuning
8	MCFE-CNN-LSTM (Proposed)	0.980	0.987	Learnable Preproc. + Temporal LSTM

Comparative Predictions Across Deep Learning Models on Test Frames
(Green text indicates correct predictions, red text indicates errors. Proposed model is bolded.)

Input Frames	Frame 1 (Background)	Frame 2 (Bad)	Frame 3 (Bad)	Frame 4 (Good)	Frame 5 (Good)
MobileNetV2 E2E	Background (1.00) ✓	Bad (1.00) ✓	Bad (0.94) ✓	Good (1.00) ✓	Good (1.00) ✓
MCFE-Net (EfficientNet)	Background (1.00) ✓	Bad (1.00) ✓	Bad (0.78) ✓	Good (1.00) ✓	Good (1.00) ✓
MCFE-Net (MobileNetV2)	Background (1.00) ✓	Bad (0.94) ✓	Good (0.67) ✗	Good (1.00) ✓	Good (1.00) ✓
Pretrained MCFE (MobileNetV2)	Background (1.00) ✓	Bad (0.95) ✓	Good (0.77) ✗	Good (1.00) ✓	Good (1.00) ✓
MCFE-CNN-LSTM (Proposed)	Background (1.00) ✓	Bad (0.99) ✓	Bad (0.99) ✓	Good (1.00) ✓	Good (1.00) ✓

Fig. 25 Representative test video frames and corresponding model predictions. The top row shows the unique input frames (true classes: Background for Frame 1, Bad for Frames 2 and 3, Good for Frames 4 and 5), and the table lists predictions and confidence scores for each deep learning model. Frame 3 is a challenging defect frame where only some configurations (including the proposed MCFE-CNN-LSTM) classify correctly.

The results reveal a clear dichotomy between the classical and deep learning models. The best classical configuration achieves a recall of 0.995 and an overall F1-score of 0.997 when evaluated on the binary background vs. active welding task. This indicates that training classical models on oversampled embedding spaces is highly effective for

coarse active state detection, but it does not solve the fine-grained 3-class classification problem. In contrast, the end-to-end deep learning models successfully adapt to the 3-class test split. Among these, the End-to-End MobileNetV2 achieves the highest overall F1 (0.991), and the proposed MCFE-CNN-LSTM achieves 0.987 F1 and a defect recall of 0.980. This demonstrates that learnable preprocessing combined with temporal recurrence matches the performance of end-to-end models while preserving structural interpretability, which is vital for industrial process validation.

7.7 Real-Time Model Evaluation Interface

To facilitate real-time monitoring and visual verification of the models' inference capabilities, we developed an interactive evaluation frontend powered by a FastAPI backend (Figure 26). This interface allows operators to select any trained model architecture (such as MobileNetV2, EfficientNet-B0, or the proposed MCFE-CNN-LSTM) and run inference on unseen welding videos. The web-based dashboard visualizes the model's accuracy, F1-scores, and per-class performance in real time. It displays a continuous temporal timeline comparing the actual vs. predicted quality states, alongside a frame viewer that shows predicted and actual labels for individual frames. This web interface serves as a practical deployment tool for verifying model generalizations and defect-detection performance in an active manufacturing environment.

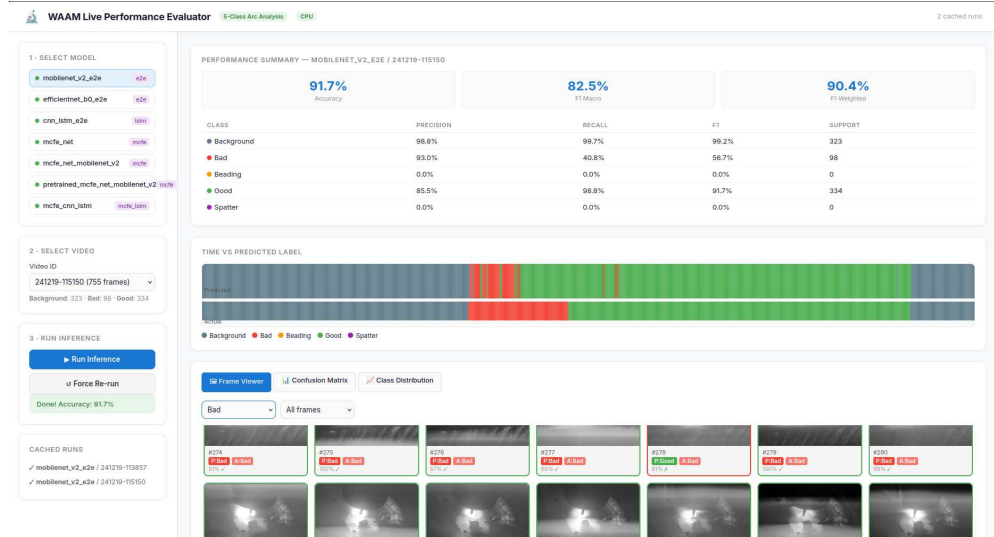


Fig. 26 Interactive WAAM Live Performance Evaluator dashboard featuring model selection, real-time metrics visualizer, actual vs. predicted timelines, and frame-by-frame quality labeling.

Specifically, the real-time classification predictions (Background, Good, Bad) can be routed to a closed-loop control system. When a "Bad" state (defect/spatter/beading) is predicted with high confidence, a negative feedback loop can automatically

reduce travel speed or lower wire feedstock input to stabilize the welding arc before the defect propagates. Conversely, during stable deposition ("Good"), parameters can be optimized for throughput.

7.8 Human-in-the-Loop Validation

In addition to quantitative and qualitative evaluation, the predictions of the proposed MCFE-Net and MCFE-CNN-LSTM frameworks were further validated through human-in-the-loop inspection using an interactive Dataset Annotation and Live Performance Dashboard to ensure the correctness and reliability of WAAM quality assessment. This process involved manual frame-by-frame examination of detected bead morphology anomalies (beading), spatter ejection events, active arc instability (bad weld state), and background idle states against the original video frames to verify classification accuracy. Particular emphasis was placed on challenging scenarios such as intense arc plasma glare, rapid melt-pool fluctuations, and overlapping defect patterns. This human-in-the-loop validation provides an additional level of confidence in the model outputs, complementing metric-based evaluation and strengthening the practical applicability, interpretability, and trustworthiness of the proposed learnable feature engineering and temporal modeling framework for real world WAAM monitoring and closed-loop parameter control.

7.9 Accuracy Evaluation

To systematically evaluate the performance of the proposed MCFE-CNN-LSTM framework against the baseline models, we analyze the multi-class classification metrics on the held-out test video sequence. The MCFE-CNN-LSTM architecture achieves a high overall F1-score of 0.987 and a defect recall of 0.980, demonstrating its robustness in identifying critical defect states (such as spatter and beading) under realistic class-imbalanced conditions. This represents a substantial improvement over the stateless EfficientNet-B0 baseline (defect recall of 0.531) and the standard CNN-LSTM (defect recall of 0.612). By coupling the learnable preprocessing modules (CEM and ESM) with temporal recurrent modeling, the network successfully suppresses transient false positives caused by flying spatter and intense arc flashes while maintaining high sensitivity to gradual weld pool geometry variations. The resulting high precision (1.00 for the defect class) is highly valuable for industrial manufacturing, as it minimizes costly false alarms and ensures that closed-loop parameter corrections are triggered only by actual deposition anomalies.

7.10 Accuracy Improvement

While the proposed framework shows near-perfect classification performance on the test split, further enhancements can be implemented to improve generalization across broader manufacturing envelopes. First, addressing the extreme class imbalance through advanced sequence-level resampling techniques, such as temporal SMOTE or Generative Adversarial Networks (GANs) tailored for video sequences, could synthesize a wider variety of rare defect modes. Second, replacing the LSTM recurrence block

with a Transformer encoder utilizing self-attention mechanisms would enable the network to capture longer temporal dependencies and multi-scale sequential interactions more effectively. Finally, integrating auxiliary physical sensor data—specifically real-time current, voltage, and wire feed speed measurements—into a multi-modal fusion architecture would provide complementary physical context to the visual features, mitigating visual ambiguities caused by severe arc glare or optical occlusion.

8 Conclusion

This paper presented a systematic, progressive evaluation of eight evaluated configurations from five distinct model families for real-time WAAM defect detection, with each architectural decision empirically justified.

Classical Two-Stage Pipeline Classifiers on frozen spatial embeddings established a strong baseline for binary Active-vs-Background detection. Under the binarized setting, the best classical configuration (CLAHE + EfficientNet-B0 + Random Forest) achieved an Active class recall of 0.995 and overall F1 of 0.997. However, fixing the backbone and preprocessing parameters exposes a fundamental information bottleneck that restricts these models to binary active state detection rather than fine-grained 3-class quality monitoring.

End-to-End Fine-Tuning Releasing all parameters to joint gradient optimization (MobileNetV2: 99% accuracy) overcomes the classical pipeline’s imbalance bottleneck, raising the Bad class F1 to 0.99. Per-frame stateless models, however, remain limited in capturing long-term defect onset dynamics.

CNN-LSTM Temporal Architecture Adding an LSTM over a 5-frame sliding window raised accuracy to 94% and achieved a Bad class F1 of 0.74. The gate mechanisms suppress single-frame arc noise, demonstrating that temporal context is the single most impactful architectural addition for WAAM defect classification.

MCFE-Net and MCFE-CNN-LSTM Replacing fixed preprocessing with learnable CEM and ESM modules unifies preprocessing and classification into a single jointly optimized system. The standalone MCFE-Net (EfficientNet backbone) yields 98% accuracy and high Bad class precision (1.00). The proposed MCFE-CNN-LSTM combines learnable preprocessing with temporal modeling, achieving 0.980 Defect Recall while providing the unique capability of visualizing learned preprocessing filters as interpretable domain knowledge.

Future work includes: (1) applying SMOTE or GAN-based synthesis to the under-represented defect class; (2) replacing the LSTM with a Transformer encoder to model longer temporal dependencies; and (3) integrating the 3-class prediction output directly into a closed-loop controller that modulates current or torch speed in real time.

Declarations

- Funding: Author Sagar Nikam acknowledges the funding support provided by the Department for Science, Innovation and Technology (DSIT), through International Science Partnership Funding (ISPF) awarded to Ulster University via the Department for the Economy (DfE), Northern Ireland.

- Conflict of interest: The authors declare no conflicts of interest.
- Ethics approval and consent to participate
- Consent for publication
- Data availability: The datasets used and analysed during the current study are available from the corresponding author on reasonable request.
- Materials availability: Not applicable
- Code availability: available from the corresponding author on reasonable request.
- Author contribution: **Soham Agarwal:** conceptualisation, formal analysis, investigation, methodology, validation, writing—original draft. **Shahana Bano:** conceptualisation, supervision, investigation, methodology, project administration, validation, writing—drafting and editing **Anil Prathuru:** conceptualisation, methodology, project administration, validation, writing—drafting and editing. **Somasundar Kannan:** formal analysis, writing—drafting and editing. **Deepika Nikam:** methodology, data acquisition, validation. **Shiva Sekar:** methodology, data acquisition, validation. **Sagar Nikam:** funding acquisition, conceptualisation, methodology, data acquisition, writing—drafting and editing.

References

- [1] Akshansh Mishra, “CLAHE Approach for Enhancement of the Microstructures of Friction Stir Welded Joints,” *Materials Today: Proceedings*, 2021.
- [2] S. W. Williams, F. Martina, A. C. Addison, J. Ding, G. Pardal, and P. Colegrove, “Wire + Arc Additive Manufacturing,” *Materials Science and Technology*, vol. 32, no. 7, pp. 641–647, 2016.
- [3] Kim, D., Lee, C., and Kim, D. (2024). Heliyon Determining optimal bead central angle by applying machine learning to wire arc additive manufacturing (WAAM). *Heliyon*, 10(1), e23372. <https://doi.org/10.1016/j.heliyon.2023.e23372>
- [4] Franke, Jan, Florian Heinrich, and Raven T. Reisch. 2025. “Vision Based Process Monitoring in Wire Arc Additive Manufacturing (WAAM).” *Journal of Intelligent Manufacturing* 36 (3): 1711–21. <https://doi.org/10.1007/s10845-023-02287-x>
- [5] Clark, Benton, Syed Ibn Mohsin, Hasan A. Poonawala, Fazleena Badurdeen, Raymond Brennan, and I. S. Jawahir. 2025. “Real-Time in-Situ Defect Detection via Multi-Modal Classification of Waveforms for Wire Arc Additive Manufacturing.” *Journal of Intelligent Manufacturing* 37 (2): 615–27. <https://doi.org/10.1007/s10845-024-02561-6>
- [6] Kim, Keun Woo, Alexander Kamerkar, Tzu-En Chiu, Ibrahim Abdi, Jian Qin, Wojciech Suder, and Seemal Asif. 2025. “WAAM-ViD: Towards Universal Vision-Based Monitoring for Wire Arc Additive Manufacturing.” *Frontiers in Manufacturing Technology* 5 (October): 1–19. <https://doi.org/10.3389/fmtec.2025.1676365>

- [7] Gadde, D., Elwany, A., and Du, Y. (2025). Deep Learning to Analyze Spatter and Melt Pool Behavior During Additive Manufacturing. *Metals*, 15(8), 1–18. <https://doi.org/10.3390/met15080840>
- [8] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A Large-Scale Hierarchical Image Database,” in *Proc. IEEE CVPR*, 2009, pp. 248–255.
- [9] S. J. Pan and Q. Yang, “A Survey on Transfer Learning,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.
- [10] M. Tan and Q. V. Le, “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks,” in *Proc. ICML*, 2019, pp. 6105–6114.
- [11] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” in *Proc. IEEE CVPR*, 2016, pp. 770–778.
- [12] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “MobileNetV2: Inverted Residuals and Linear Bottlenecks,” in *Proc. IEEE CVPR*, 2018, pp. 4510–4520.
- [13] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” in *Proc. ICLR*, 2015.
- [14] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [15] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-C. Woo, “Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting,” in *Advances in Neural Information Processing Systems*, 2015, pp. 802–810.
- [16] Y. Zhao, J. Ye, and Z. Li, “A Deep Learning Based Visual Detection Method for WAAM Process Monitoring,” *Journal of Manufacturing Processes*, vol. 56, pp. 985–993, 2020.